System Programming & OS 실습

0. Introduction

이성현, 최민국

Dankook University

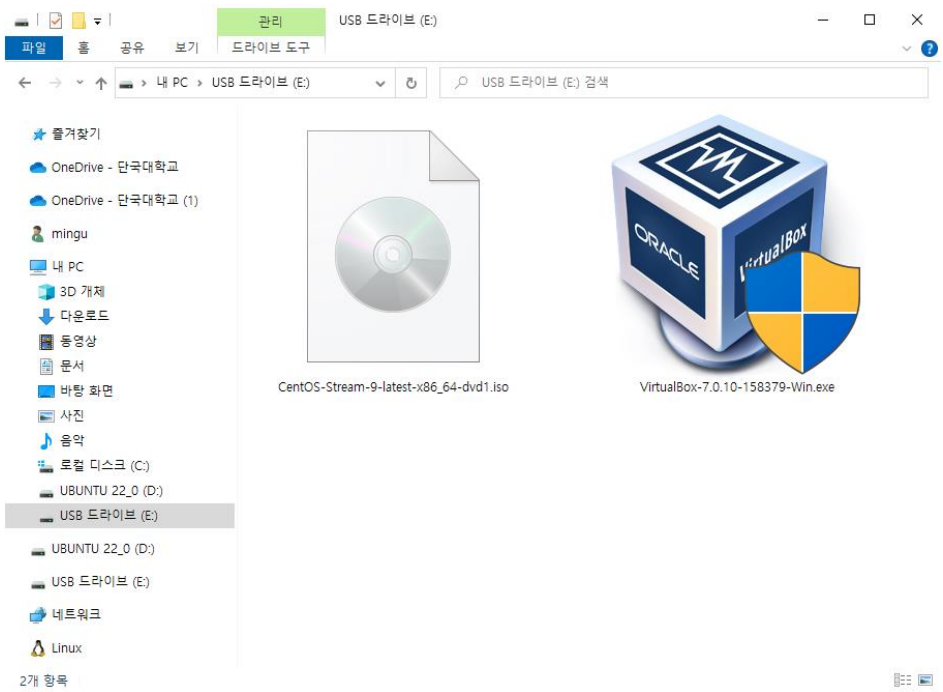
{leesh812, mgchoi}@dankook.ac.kr

Teaching Assistant

이성현
단국대 인공지능 융합대학 박사과정
시스템 소프트웨어 연구실(최종무 교수님)
leesh812@dankook.ac.kr

최민국
단국대 인공지능 융합대학 석사과정
시스템 소프트웨어 연구실(최종무 교수님)
mgchoi@dankook.ac.kr

Before we start...



USB에 있는 아래 2가지파일 실습 컴퓨터에 복사해주세요.
마지막으로 복사한 학생이 조교에게 반납해주세요.

VirtualBox-7.0.10-158379-Win.exe
CentOS-Stream-9-latest-x86_64-dvd1.iso

What is System Program?

System software

44 languages

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

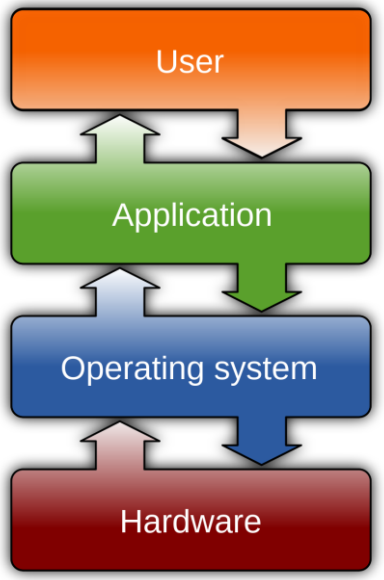
System software is software designed to provide a platform for other software. Examples of system software include operating systems (OS) (like macOS, Linux, Android, and Microsoft Windows), computational science software, game engines, search engines, industrial automation, and software as a service applications.[1]

Operating system

Article Talk

From Wikipedia, the free encyclopedia

An **operating system** (**OS**) is system software that manages computer hardware and software resources, and provides common services for computer programs.



What is System Program?

- Software components
  - ✓ Application program vs. System program

```
#include <stdio.h>

int main()
{
    printf("Hello, World₩n");
}
```

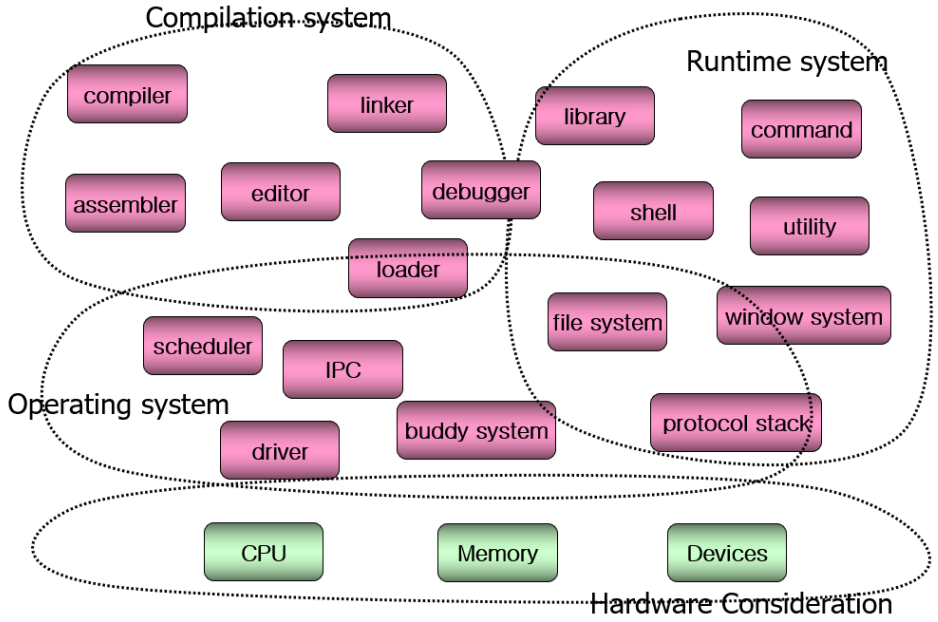
- How to run this program on CPU?
- What is the role of printf()?
- How the string is displayed on Monitor?
- What are the differences between local and global variables?
- What kinds of techniques can be applied to enhance the performance of this program?

What is System Program?

- Software components: System program
  - ✓ How to run this program on CPU?
    - *Object, binary, compiler, assembler, loader, ...*
  - ✓ What is the role of printf()?
    - *Library, linker, ...*
  - ✓ How the string is displayed on Monitor?
    - *Device driver, file system, ...*
  - ✓ How a program can be executed with other programs concurrently?
    - *Scheduler, context switch, IPC(Inter process communication), ...*
  - ✓ What are the differences between local and global variables?
    - *Data, stack, heap, virtual memory, buddy system, ...*
  - ✓ What kinds of techniques can be applied to enhance the performance of this program?
    - *Compiler optimization(loop unrolling, reordering), CPU optimization(pipeline, superscalar, out-of-order execution)*

What is System Program?

- Software components: System program
  - ✓ Supporting computing environments for application programs
    - Support Interfaces such as commands, library functions and system calls
  - ✓ Strongly related to hardware(hardware management)



Why should we learn System Program?

LINE

2022하반기신입 LINER공개채용

우리 조직은 LINE 서비스에서 필요로 하는 프론트엔드 개발 및 관련 도구를 개발하고 운영합니다.
각기 다른 국가의 환경에 맞춰 최적의 기술 스택을 직접 고민하고 선정하여 개발 환경을 구축할 뿐 아니라,
유저의 사용성을 높이기 위한 기술을 연구하고, 이에 대한 의견을 적극적으로 반영하며
성능을 개선하기 위한 최적화에 대해 끊임없이 고민하고 노력합니다.

[신입 LINER가 되시면 이런 업무를 담당할 수 있습니다]
- LINE의 웹서비스를 위한 프론트 서버 및 클라이언트 개발
- LINE의 Social Service인 LINE VOOM 서비스 운영 및 신규 개발

Global Web Software Engineer (Front-end)

[신입 LINER에게 바랍니다]
- 컴퓨터 과학의 기초 지식을 이해하고 활용할 수 있으신 분 (자료 구조, 알고리즘, 데이터베이스, 네트워크, 운영 체제 등)
- JavaScript, HTML, CSS를 다루며 관련 내부 동작 원리에 대한 관심이 있으신 분
- Git을 이용한 코드 형상 관리와 코드 리뷰 경험이 있으신 분
- 해외 출장 및 근무에 결격 사유 없으신 분

[필수는 아니지만 아래의 경험이 있으면 좋아요]
- React, Vue.js, Angular 등 MV*기반의 Framework를 이용한 실무 경험이 있으신 분이면 좋아요
- Babel를 이용한 ES6+ 스펙의 사용 경험이 있으신 분이면 좋아요

※ 최종 합격 시, LINE PLUS 법인으로 입사하시게 됩니다. (근무지: 분당구 서현동 분당스퀘어)

tech-interview-for-developer

since 2019.03.01 | author gyoogle | license MIT | hits 566 / 932527 | all contributors 58 | PRs welcome

Watchers 151 | Stars 12k | Forks 2.9k

- Operating System
  - 운영체제란
  - 프로세스 vs 스레드
  - 프로세스 주소 공간
  - 인터럽트(Interrupt)
  - 시스템 콜(System Call)
  - PCB와 Context Switching
  - IPC(Inter Process Communication)
  - CPU 스케줄링
  - 데드락(DeadLock)
  - Race Condition
  - 세마포어(Semaphore) & 뮤텍스(Mutex)
  - 페이징 & 세그먼테이션 (PDF)
  - 페이지 교체 알고리즘
  - 메모리(Memory)
  - 파일 시스템
- Computer Architecture
  - 컴퓨터 구조 기초
  - 컴퓨터의 구성
  - 중앙처리장치(CPU) 작동 원리
  - 캐시 메모리
  - 고정 소수점 & 부동 소수점
  - 패리티 비트 & 해밍 코드
  - ARM 프로세서

https://github.com/gyoogle/tech-interview-for-developer

Why are hands-on lessons necessary?

Experience is the best teacher, and the worst experiences teach the best lessons.

Lecture Schedule

- 1주차
  - ✓ 오전 : 09:00 ~ 10:00 [고정]
  - ✓ 오후
    - 9/4(월) : 13:30 ~ 14:30
    - 9/5(화) : 15:30 ~ 16:30
    - 9/7-8(목/금) : 13:30 ~ 14:30

- 2주차
  - ✓ 오전 : 09:00 ~ 12:00 [고정]
  - ✓ 오후
    - 9/11-12(월/화) : 13:30 ~ 18:00
    - 9/14(목) : 13:30 ~ 16:00 ** 16시 부터 교수님 Q&A
    - 9/15(금) : 13:30 ~ 14:30 ** 14시30분 부터 교수님 Q&A, 16시 부터 시험

- 수업 일정은 변경될 수 있음. 변경될 시 공지할 예정

Lecture Schedule

월	화	수	목	금	토	일
9/4 0. Introduction 1. Installation	9/5 2. Linux 3. SSH	9/6	9/7 4. Vim 5. Gcc	9/8 6. Synchronization 7. File I/O	9/9	9/10
9/11 9. File Programming	9/12 13. Task Programming	9/13	9/14 16:00 교수님 Q&A 10. Concurrency	9/15 14:30 교수님 Q&A 16:00 시험 11. Cloud computing 12. Docker	9/16 (온라인) * 시험해설 강의 * 8. Vim & Shell Script	9/17

Exam

- 시험
  - ✓ 9/15(금) 16:00
  - ✓ 이론 시험과 별개로, 실습 수업 시험 예정

- 시험 범위
  - ✓ 9/14(목) 강의한 내용까지
  - ✓ 수업 시간에 다루지 않은 내용은 출제 하지 않음

- 강의 PPT
  - ✓ https://github.com/DKU-EmbeddedSystem-Lab/TABA_OS_2023

Q&A